



Whitehouse, S., Yordanova, K., Paiement, A., & Mirmehdi, M. (2017). Recognition of unscripted kitchen activities and eating behaviour for health monitoring. In *2nd IET International Conference on Technologies for Active and Assisted Living (TechAAL 2016): Proceedings of a meeting held 24-25 October 2016, London, UK* (pp. 1-6). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1049/ic.2016.0050>

Peer reviewed version

License (if available):
Unspecified

Link to published version (if available):
[10.1049/ic.2016.0050](https://doi.org/10.1049/ic.2016.0050)

[Link to publication record in Explore Bristol Research](#)
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at <http://ieeexplore.ieee.org/document/7801334/>. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Recognition of unscripted kitchen activities and eating behaviour for health monitoring

Samuel Whitehouse*, Kristina Yordanova⁺, Adeline Paiement*, Majid Mirmehdi*

*University of Bristol, UK – sw12690@my.bristol.ac.uk, csatmp@bristol.ac.uk, majid@cs.bris.ac.uk

⁺University of Rostock, Germany – kristina.yordanova@uni-rostock.de

Keywords: Activity recognition, Intention recognition, Kitchen activities, Health monitoring, Unscripted activities

Abstract

Nutrition related health conditions such as diabetes and obesity can seriously impact quality of life for those who are affected by them. A system able to monitor kitchen activities and patients' eating behaviours could provide clinicians with important information helping them to improve patients' treatments. We propose a symbolic model able to describe unscripted kitchen activities and eating habits of people in home settings. This model consists of an ontology which describes the problem domain, and a Computational State Space Model (CSSM) which is able to reason in a probabilistic manner about a subject's actions, goals, and causes of any problems during task execution. To validate our model we recorded 15 unscripted kitchen activities involving 9 subjects, with the video data being annotated according to the proposed ontology schemata. We then evaluated the model's ability to recognise activities and potential goals from action sequences by simulating noisy observations from the annotations. The results showed that our model is able to recognise kitchen activities with an average accuracy of 80% when using specialised models, and with an average accuracy of 40% when using the general model.

1 Introduction

Proper nutrition is an effective and cheap way to improve both quality and longevity of human life, since it lowers risk factors associated with nutrition related diseases [9]. This is particularly true for physical conditions, such as diabetes, or mental conditions, such as depression, that affect a patient's willingness to prepare and consume healthy food. People suffering from dementia also often have poor nutrition since their ability to prepare food is impaired by their disease [14].

To reduce costs associated with hospitalisation and treatment of these conditions, some research has attempted to automate home monitoring for patients. This can also improve general well-being as patients can be monitored and treated in home settings [10]. Since such systems are designed to run without the intervention of medical personnel or the patients themselves, costs associated with frequent doctors visits or patients unable to self-manage their treatments are reduced. Composite information compiled from environmental sensor

data ensures the delivery of relevant information accessible to healthcare professionals, designated carers and/or family.

For more focused tasks, such as monitoring quality of eating movement in recovering stroke victims, or calorie intake from images of meals, statistical analysis and simple classifiers would be appropriate, as in [12] and [2] respectively. However, more complex systems relevant to a broader range of medical applications would require the ability to understand some aspects of human behaviour.

Activity Recognition (AR) is a key requirement of such systems as it allows for machine understanding of relevant human actions. Using sensor data, it allows a machine to reason about the actions a human performs and therefore estimate the current state of the world. Such a model could also look at whole or partial sequences of actions and predict the patients' goals and the methods by which they achieve those goals. This can be monitored over time to detect changes in their habits, which in turn can be used to detect deviations in behaviour indicating progression of a medical condition.

There are two main AR paradigms: data-driven and knowledge-based (or context aware) [16]. Data-driven approaches rely on large datasets from which a model is learnt, with additional modifications afterwards to better fit a specific purpose. They can be either unsupervised (i.e. clustering) [3], or supervised (i.e. classification based on training data) [15]. Data-driven approaches suffer from two main limitations: they need large quantities of sensor data to improve performance, and they are limited to behaviours present in the training data.

To address these problems, knowledge-based approaches rely on domain knowledge in the form of symbolic models and rules to reason about observed behaviours [11, 7]. These approaches are advantageous since they are able to reason beyond sensor data and provide information about the patient's situation and potential medical reasons for their behaviour [14]. The main challenge for knowledge-based approaches is a common inability to cope with problems associated with real world scenarios, such as variability of user behaviour resulting in computationally infeasible models and imperfect sensors leading to recognition ambiguity in purely symbolic models.

In order to meet these challenges, we propose the use of Computational State Space Models (CSSMs) [4] which combine symbolic representation with probabilistic reasoning to compensate for behaviour variability and sensor noise. So far, CSSMs have only been applied to scripted scenarios in simpli-

fied settings that do not address the challenges of complexity and behaviour variability present in real settings.

In this work we investigate the application of CSSMs to unscripted kitchen scenarios and the corresponding domain knowledge representation. To that end, we model the cooking and eating behaviour of different people preparing unscripted meals in a living lab. We also make publicly available the ontologies, annotations and simulated sensor data, which are useful for evaluation of future AR algorithms and provide new insights into the complexity of unscripted kitchen activities.

Section 2 presents current methods of knowledge-based activity recognition. Section 3 gives an overview of the development process and the tools used for creation of the CSSM. Section 4 outlines the models and ontologies developed. Section 5 evaluates the performance of the models. Section 6 concludes the paper and suggests future work.

2 Related Work

There are a variety of approaches for knowledge-based activity recognition that allow incorporation of context information into a model. One common paradigm features libraries of plans which are explicitly provided by human experts. For example, Roy et al. [7] relies on manually created ontology-based plan libraries. These plans represent partially ordered sequences of actions that must be carried out in order to achieve a goal. As Yordanova and Kirste [18] point out, “library-based models are inherently unable to solve the problem of library completeness caused by the inability of a designer to model all possible execution sequences leading to the goal”.

An alternative option for arriving at a suitable model is to mine action sequences from observations of human behaviour. For example, Chen et al. [1] use an ontology-based approach to manually define an initial library of behaviours. Later, observations of user activities are used to add variations or remove obsolete behaviours. Although the approach provides an interesting solution to the problem of keeping plan libraries up-to-date, it still relies on initial manual definitions of behaviours.

To address this problem, Ye et al. [16] proposes an approach similar to that of Chen et al. [1]. They replaced the initial ontology definition with an ontology learned from textual sources and utilised unsupervised sensor segmentation and learning to build their library of behaviours. This approach suffers from similar problems to data driven approaches as it only learns observed behaviours, leading to a behaviour variability which is dependant on the amount of training data.

In order to avoid the problem of low behavioural variability without relying on large amounts of sensor data, Computational State Space Models (CSSMs) were investigated [11, 13, 4, 18]. CSSMs describe actions in terms of preconditions and effects, allowing probabilistic reasoning about user states, goals and context. This manually defined model is very compact as it only contains the definition of basic action templates which are automatically expanded into different execution sequences based on their causal relationships. This provides an alternative solution to the problem of manually defining all execution sequences, or mining them from large

amounts of annotated sensor data.

3 Methods and Materials

We now describe the formalisms used to model our problem, the development process we followed, the experimental settings, and the collected sensor data.

3.1 Computational Causal Behaviour Models

We selected Computational Causal Behaviour Model (CCBM) as the tool for creation of our CSSMs as it has been shown to perform accurately with large state spaces and imperfect observations [4].

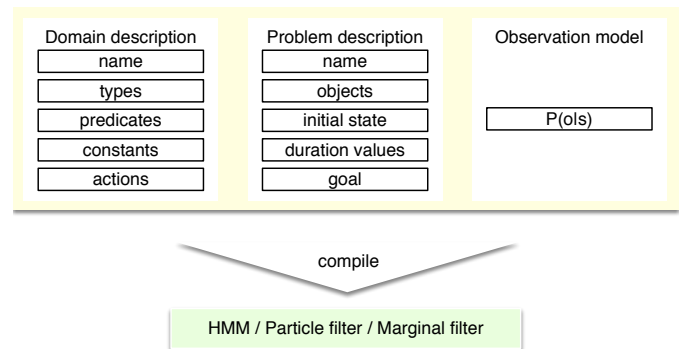


Figure 1. Elements of a Computational Causal Behaviour Model. Figure adapted from [17].

Figure 1 shows the structure of a CCBM, consisting of domain and problem definitions as well as an observation model. The domain definition contains the set of possible actions expressed in terms of precondition-effect rules. Preconditions define what has to be true in order for the action to take place, while effects describe how the execution of an action changes the state of the world. The problem definition defines which elements of the environment are available (e.g. objects, locations, persons, etc.), the initial condition or state of the environment and possible goals that could be achieved in this problem. Finally, the last part of the model is an observation model that gives the probability of observing a certain sensor value, given the current state. To develop our CSSM, we make use of the Computational Causal Behaviour Modelling (CCBM) tool [19], which allows for the definition of the three model elements. These are then automatically translated by the tool into a probabilistic inference model (e.g. hidden Markov model, particle filter, marginal filter) that is capable of estimating the current state of the world along with predictions about potential future states or goals. For more details about CCBM, see the paper by Yordanova and Kirste [18].

3.2 A Process for developing CSSMs

To develop a CSSM we follow the process described in Yordanova and Kirste [18] as illustrated in Figure 2. This begins with the analysis phase where the problem to be modelled is analysed and sets of possible actions and relevant environment

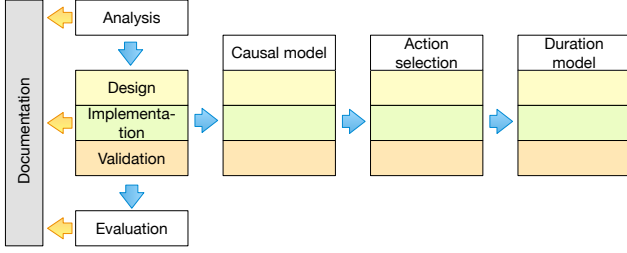


Figure 2. Development process for CSSMs as proposed in [18].

elements are selected. Later, sensor data that is used to test the model to be developed is collected. In this phase, an ontology is created that incorporates the context information about the problem to be modelled. The collected sensor data is then annotated based on the action schema described in the ontology.

Development of the causal model constitutes the second phase, in which the symbolic model needs to be designed, implemented and validated. These are the rules describing the actions in terms of preconditions and effects as well as the problem description, as shown in Figure 1. The third phase covers development of the action selection heuristics, with an emphasis on action preference dependant on a given state. This phase mainly consists of design, implementation, and validation of these heuristics. The fourth phase is the development of the actions’ durations, represented in terms of probability distribution, including the decision about which type of probability distribution to use. Finally, the developed model is evaluated based on the collected sensor dataset to see if it is able to perform activity recognition.

In this work, we concentrate on the problem analysis and the development of the causal model, beginning with modelling the symbolic rules and observation model in the CCBM notation and then validating them. The validation consists of generating plans (descriptions of action sequences) from the annotated data and checking whether the model is able to explain them. This can be done both with taking action durations into account or without, and is useful for highlighting potential issues in the model earlier in the project. We used noisy simulated observations generated from the annotation sequences for this validation, without taking the action durations into account.

3.3 Experimental design

In order to evaluate the model, a dataset describing cooking and eating behaviour was collected in the SPHERE House. The SPHERE project (a Sensor Platform for HEalthcare in a Residential Environment) is an interdisciplinary research project with the remit to provide an in-home monitoring solution to assist medical professionals in providing care for their patients [6]. The SPHERE House in Bristol (UK) is a typical terraced residence which is used as a living lab for experimentation on sensors and systems in a realistic environment. The house itself is deliberately normal and the systems within have been installed as they would be in a patient’s residence, forcing de-

sign considerations which would be standard for a functioning dwelling.

The sensor network in the SPHERE kitchen collects data on *temperature, humidity, light levels, noise levels, dust levels, motion within the room, cupboard and room door state and water and electricity usage*. There is also an RGB-D (depth sensing) camera in the room which is used to provide positional data on the occupants [5]. A head-mounted camera was used to record the actions of the participants during the study to allow for better annotation of the observations.

The collected dataset contains sequences of individual human protagonists performing varied and complex activities in the SPHERE kitchen, without any predefined scripts. Thus, the dataset is a good example of natural human behaviour and highlights some key issues which need addressing by models attempting to recognise “real” human activities, such as irrational behaviours, multitasking, and working towards two or more goals at the same time.

Each data collection event took place over the course of around two hours including breaks in the kitchen of the SPHERE house, involving 9 participants. The only instructions they received were to prepare a meal and/or a drink of their choice in the kitchen. This resulted in the collection of 15 unscripted meal preparation and consumption tasks. The meals/drinks included: pasta, ready meal, carrot sticks, rice and vegetables, toast, juice, tea, coffee, chicken and vegetables, snack, rice and curry, macaroons, salad, and toasted cheese sandwiches. A total of 449 minutes were recorded with individual recording durations between 10 and 88 minutes.

4 Developed Model

Below we provide information about our reasoning when modelling the domain knowledge. We also present the resulting ontologies, data annotation and the CCBM model. The ontologies and the simulated data are publicly available¹.

4.1 Ontologies

The ontology represents the set of actions that can be executed in our problem, locations where they can be executed, and objects in the environment relating to their execution. When designing our models, we considered two different ontologies which could be compared to each other in order to examine how different levels of detail could be applied to the problem. For the first, we made a fine-grained ontology describing each action with a high level of detail, and for the second, a coarse-grained ontology describing a more general overview of actions being performed without consideration for finer details.

Fine-grained ontology – The fine-grained ontology covers a variety of potential kitchen actions and as such is reasonably exhaustive, with *prepare* and *prepare-meal* being used to make sense of many different preparation actions, such as removing packaging, stirring, and chopping. Action schemata for this ontology are shown in Table 1. Subjects of these actions (i.e. *location, cupboard, appliance, item, counter, food, container*)

¹<https://data.bris.ac.uk/data/dataset/raqa2qzai45z15b4n0za94toi>

- 1) *move* - < location > - < location >
- 2) *open* - < cupboard >
- 3) *close* - < cupboard >
- 4) *turnon* - < appliance >
- 5) *turnoff* - < appliance >
- 6) *wash* - < item >
- 7) *clean* - < counter >
- 8) *drink* - < food >
- 9) *eat* - < food >
- 10) *take* - < item > - < container >
- 11) *put* - < item > - < container >
- 12) *prepare* - < food > - < counter >
- 13) *prepare - meal* - < counter >

Table 1. The actions schemata for the first iteration of the fine-grained ontology.

are sets of objects included within the experimental space. The names of these sets are self-descriptive with some overlap between sets, for example the oven is both a cupboard (things can be put into it) and an appliance (it can be turned on and off).

The *move*, *open*, *close*, *turnon* and *turnoff* actions were included in the ontology because they roughly align with the sensor data available (cupboard door sensors, power usage sensors, and position tracking sensors). *wash*, *clean*, *drink* and *eat* relate to potential goals to be considered, such as whether washing up had been done and whether food or drink had been consumed. Finally, *take*, *put*, *prepare* and *prepare-meal* are all actions which a protagonist performs in pursuit of the above goals, such as moving items around the kitchen to the correct positions and preparing food and drink.

Revised fine-grained ontology – While developing the fine-grained model, we determined that the fine-grained action schemata and corresponding objects ontology was too complex, resulting in many possible states when executing an action. This in turn reduced the probability of each potential action to almost zero. Hence, we revised the fine-grained ontology to strike a better balance between ease of annotation, computability, and model complexity. The modified version with the new action schemata is shown in Table 2.

- 1) *move* - < location > - < location >
- 2) *open* - < cupboard >
- 3) *close* - < cupboard >
- 4) *turnon* - < appliance >
- 5) *prepare - meal* - < location >
- 6) *prepare* - < food > - < location >
- 7) *take* - < item > - < location >
- 8) *put* - < item > - < location >
- 9) *eat* - < food >
- 10) *clean*
- 11) *drink* - < food >
- 12) *wash* - < food >
- 13) *turnoff* - < appliance >
- 14) *dispose - waste*
- 15) *wash - hands*
- 16) *wash - items*

Table 2. The actions schemata for the second iteration of the fine-grained ontology.

Included are three additional composite actions to describe common tasks, as well as revisions for other actions to reduce the number of object sets. These sets are still descriptive of items in the kitchen, but with less differentiation between items of similar types leading to better model performance (see Table 3)². One key removal was the idea of containers, which were leading to expensive nested searches for items.

Coarse-grained ontology – The coarse-grained ontology is simpler by design, and aims to represent the bare minimum information needed to recognise the type of meal and cooking

²Note that grouping items does not reduce the CCBM granularity as activities are still executed in the same way. The difference is a reduction in complexity, as instead of tracking n distinct instances of the same object, we now track the presence of a type of object (e.g. an action is performed using any knife in the area).

Location	bin, corridor, counterkettle, counterleft, countermiddle, counterright, counterrightback, cupboardleft, cupboardright, cupboardtopleft, cupboardtopright, drawbottom, drawmiddle, drawtop, dryingrack, floormiddle, fridge, hob, oven, self, sink
Item	bread, clean-tool, container, cup, dishware, fruit, implement, juice, kettle, meat, milk, oil, pasta, readymeal, sauce, snack, spices, vegetables, waste, water
Food	bread, fruit, juice, meat, milk, oil, pasta, readymeal, sauce, snack, spices, vegetables, waste, water
Cupboard	counterrightback, cupboardleft, cupboardright, cupboardtopleft, cupboardtopright, drawbottom, drawmiddle, drawtop, fridge, microwave, oven
Appliance	hob, kettle, microwave, oven, tap, toaster

Table 3. Object sets for the fine-grained ontology.

activity. This ontology is more goal oriented, with actions referring to the meal they are contributing towards instead of the process by which they are being completed. The action schema for this ontology is shown in Table 4.

- 1) *move* - < location > - < location >
- 2) *get* - < item > - < meal >
- 3) *put* - < item > - < meal >
- 4) *prepare* - < meal >
- 5) *eat* - < meal >
- 6) *drink* - < meal >
- 7) *clean*

Table 4. The actions schemata for the coarse-grained ontology.

Again, *location* and *item* are sets of objects included within the experimental space, with *meal* referring to the eventual goal. With a reduced number of different sets in this ontology, resulting models are much simpler and quicker to compute.

The *move*, *eat* and *drink* actions are similar to their equivalents in the fine-grained model, while *get*, *put*, and *prepare* have been changed to focus less on location and more on the goal of the food making process. *clean* no longer has an argument, since for this ontology we are more interested in whether cleaning has been done than what has been cleaned.

Both ontologies have an irrational action, *unknown*, which is used in situations when the current action being performed is not relevant when considering potential goals.

Figure 3 shows the object ontology for the coarse-grained action schemata. Rectangles represent concrete objects while ellipses represent types to which objects belong. These types correspond to types in the action schemata; for example in *move* - < location > - < location >, “location” represents the type for all locations, including the concrete instances “kitchen” and “study” (see Figure 3).

4.2 Data annotation

The next step in the model creation process is to annotate the observation data which can be used as ground truth for evaluating the developed model. For the annotation process we used ELAN [8], a free annotation tool. For the fine-grained annotation, the action schema resulted in action sequences that comprised 24 to 676 execution steps. For the coarse-grained annotation, the action schemata resulted in considerably shorter and less complex action sequences with the shortest being 8 execution steps and the longest 199 steps.

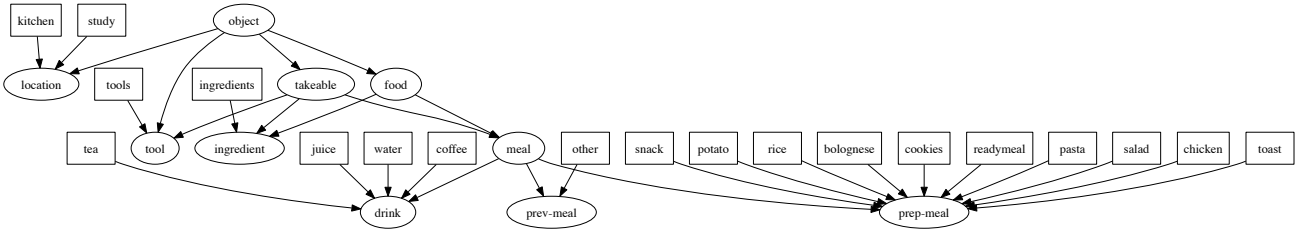


Figure 3. Coarse-grained objects ontology. Rectangles represent the concrete objects while ellipses the object types.

Parameters	Fine general	Coarse general	Coarse specialised
People	1	1	1
Locations	21	2	2
Objects	23	17	3–6
Action types	15	8	8
Action templates	27	8	8
Ground actions	442	92	10 – 28
States	$> 10^8$	450144	40 – 1288
Valid plans	$> 10^8$	21889393	162 – 15689

Table 5. Parameters for the different models.

4.3 Causal Models

Using the ontologies developed, the CCBM tool was used to implement several models: one fine-grained model using a single problem and domain file, and multiple coarse-grained models each using the same problem file but with different domain files. These were comprised of one general model which could handle all coarse-grained sequences, and one specialised coarse-grained model for each individual sequence. The model dimensions for the different implementations are in Table 5.

From this we can see that the fine-grained model is much more complex than the coarse-grained models, as a result of containing more context information about actions and elements in the environment. This information was mostly omitted in the coarse-grained model, with only the meal being prepared and the action being executed remaining.

Table 5 also shows that the specialised coarse-grained models are much simpler than the general model, with smaller state spaces due to the omission of unused objects focusing the model on the specific meal being prepared. This reduces the specialised models' ability to explain behaviour variability, but increases the model performance in terms of recognition rate as can be seen from its smaller log likelihood in Table 6.

5 Activity recognition results

To evaluate the models, we performed activity recognition with noisy simulated data. This was generated by assigning the observation matching the current annotation a value of 0.7, with all other potential observations set to 0.3. Table 6 shows the results of activity recognition on this data in terms of average branching factor (i.e. average number of actions which could

Model	Plan Length	Branching Factor	Log Likelihood
Fine-grained (g)	149	43.13	-542.42
Coarse-grained (g)	58	65.82	-239.90
Coarse-grained (s)	58	11.12	-144.48

Table 6. Average model performance statistics. “g” indicates the generalised model, “s” the specialised.

potentially be executed from each state) and the final log likelihood (i.e. likelihood of correct description of the observations).

As shown, the fine-grained model has a larger branching factor than the specialised coarse-grained models, and a lower log likelihood indicating poorer performance when explaining the observed data. This is partly due to the larger average plan length for the fine-grained model, resulting from its more atomic actions which reduce log likelihood as each actions adds uncertainty to the final state. We expect log likelihood to increase when action durations are introduced into the models.

Table 6 also shows that the general coarse-grained model has a higher branching factor than the fine-grained model. This is because the potential actions for the coarse-grained model were less constrained, since concepts such as specific locations and availability were not included.

Furthermore, it can be seen that the general coarse-grained model also has a much higher branching factor than the specialised models. The log likelihood is also affected, indicating that the specialised models explain the observations much better than the general model. Figure 4 shows the average ac-

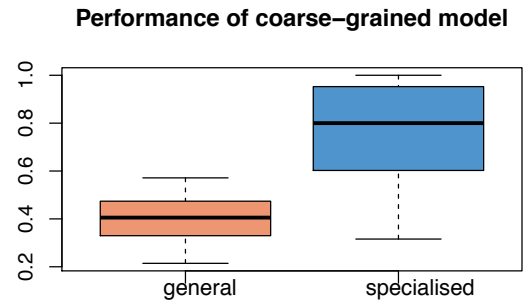


Figure 4. Performance of general and specialised coarse-grained models. The data has an error probability of 0.3.

curacy of the specialised coarse-grained model as 80%, with the accuracy of the general model only 40%. This difference in performance is due to the specialised models following only one goal, significantly reducing the number of possible actions that can be executed and therefore increasing accuracy.

6 Conclusion and Future Work

In this work we presented a CSSM for the recognition of unscripted kitchen activities in a real home. The recognition of such activities can provide better monitoring of nutrition related health conditions and thus potentially improve quality of life for patients in their own homes. Furthermore, we have made the model ontologies, simulated data and corresponding annotations publicly available. These contributions can provide important insights for the community into the complexity and behaviour dynamics of natural kitchen activities as well as test data for developing future activity recognition algorithms.

The next step in our work is the creation of an action duration model and evaluation of the models based on the actual sensor data. The ontologies for both the coarse and the fine-grained models need re-examining to cover a wider range of kitchen related activities in an efficient manner. At this time, it would be prudent to evaluate our model in comparison to alternative approaches in terms of accuracy of action identification and the overall complexity of the model.

Additional testing of the relevance of different sensors within the SPHERE kitchen would allow for improvements to the accuracy of the model, while adding new sensor groups (such as accelerometers) could improve the model performance. Collection of data from other kitchens would validate the effectiveness of the models in different settings.

Acknowledgements

We thank the SPHERE Project (ESPRC grant EP/KO31910/1) for funding this research; Sion Hannuna, Jake Hall and Przemek Woznowski for assistance with the technical details of the sensor system in the SPHERE house; and Martin Nyolt, Frank Krüger, Thomas Kirste and Max Schröder for their insights into the model building process and their CCBM tool.

References

- [1] L. Chen, C. Nugent, and G. Okeyo. An ontology-based hybrid approach to activity modelling for smart homes. *IEEE Trans. Human-Mach. Syst.*, 44(1):92–105, 2014.
- [2] A. Myers et al. Im2Calories: towards an automated mobile vision food diary. In *ICCV*, 2015.
- [3] D. Trabelsi et al. An unsupervised approach for automatic activity recognition based on hidden markov model regression. *IEEE Trans. Autom. Sci. Eng.*, 10(3):829–835, 2013.
- [4] F. Krüger et al. Computational State Space Models for activity and intention recognition. A feasibility study. *PLoS ONE*, 9(11), 2014.
- [5] J. Hall et al. Designing a video monitoring system for AAL applications: The SPHERE case study. In *TechAAL*, 2016.
- [6] N. Zhu et al. Bridging e-health and the internet of things: The SPHERE project. *IEEE Intell. Syst.*, 30(4):39–46, 2015.
- [7] P. Roy et al. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer’s patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. 2011.
- [8] P. Wittenburg et al. ELAN: a professional framework for multimodality research. In *LREC*, 2006.
- [9] S. Ohlhorst et al. Nutrition research to affect food and a healthy lifespan. *Advances in Nutrition: An International Review Journal*, 4:579–584, 2013.
- [10] A. Helal, D. J. Cook, and M. Schmalz. Smart home-based health platform for behavioural monitoring and alteration of diabetes patients. *Journal of Diabetes Science and Technology*, 3(1):141–148, 2009.
- [11] L. Hiatt, A. Harrison, and J. Trafton. Accommodating human variability in human-robot teams through theory of mind. In *IJCAI*, pages 2066–2071, 2011.
- [12] H. Hondori, M. Khademi, and C. Lopes. Monitoring intake gestures using sensor fusion (Microsoft Kinect and inertial sensors) for smart home tele-rehab setting. In *IEEE EMBS - Healthcare Innovation*, 2012.
- [13] M. Ramirez and H. Geffner. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In *IJCAI*, pages 2009–2014, 2011.
- [14] A. Serna, H. Pigot, and V. Rialle. Modeling the progression of alzheimer’s disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction*, 17(4):415–438, 2007.
- [15] E. Tapia, S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, pages 158–175, 2004.
- [16] J. Ye, G. Stevenson, and S. Dobson. Usmart: An unsupervised semantic mining activity recognition technique. *ACM TiS*, 4(4):16:1–16:27, 2014.
- [17] K. Yordanova. *Methods for engineering symbolic human behaviour models for activity recognition*. PhD thesis, Institute of Computer Science, Rostock, Germany, 2014.
- [18] K. Yordanova and T. Kirste. A process for systematic development of symbolic models for activity recognition. *ACM TiS*, 5(4):20:1–20:35, 2015.
- [19] K. Yordanova, F. Krüger, and T. Kirste. Tool support for activity recognition with computational causal behaviour models. In *German Conference on Artificial Intelligence*, pages 561–573, 2012.